PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/56

A1

(11) International Publication Number: WO 95/29549

(43) International Publication Date: 2 November 1995 (02.11.95)

(21) International Application Number: PCT/US95/04746

(22) International Filing Date: 18 April 1995 (18.04.95)

(30) Priority Data:
08/230,371 20 April 1994 (20.04.94) US

(71) Applicant (*for all designated States except US*): APPLE COMPUTER, INC. [US/US]; One Infinite Loop, Cupertino, CA 95014 (US).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): RIDDLE, Guy, Gregory [US/US]; 18243 Knuth Road, Los Gatos, CA 95030 (US).

(74) Agent: PETERSON, James, W.; Burns, Doane, Swecker & Mathis, P.O. Box 1404, Alexandria, VA 22313-1404 (US).

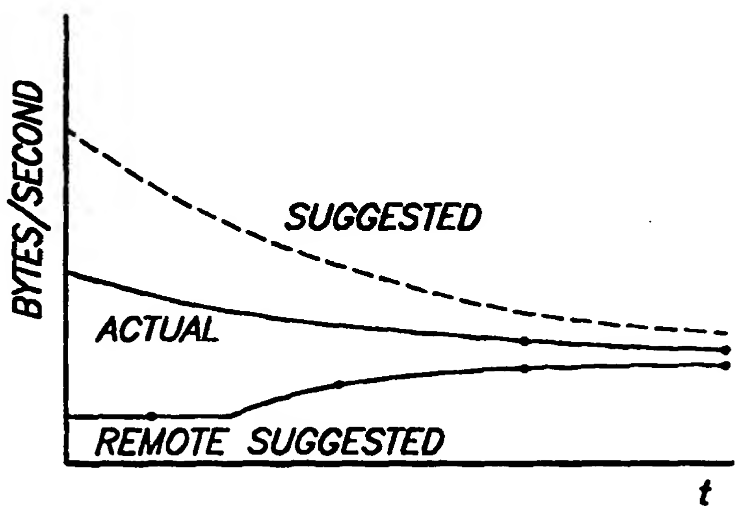
(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TT, UA, UG, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ, UG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: FLOW CONTROL FOR REAL-TIME DATA STREAMS

(57) Abstract

The present invention, generally speaking, provides for transmission and flow control of real-time data streams over computer networks. A real-time data stream is transmitted in data packets from a data source in accordance with a predetermined protocol over a shared network, for example. Data packets of said real-time data stream are received at a data destination connected to the local area network. The data destination determines a suggested data rate for the data source based in part on a number of data packets lost during a preceding interval of time and transmits the suggested data rate to the data source. The suggested data rate is received at the data source, which adjusts its data rate in accordance with the suggested data rate. The rate adjustment mechanism is designed such that a network segment will not be overloaded with a single isochronous data stream connection and that a disproportionate shared of the network bandwidth is not consumed by the isochronous data stream connection.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgystan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SI	Slovenia
CI	Côte d'Ivoire	KZ	Kazakhstan	SK	Slovakia
CM	Cameroon	LI	Liechtenstein	SN	Senegal
CN	China	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
CZ	Czech Republic	LV	Latvia	TJ	Tajikistan
DE	Germany	MC	Monaco	TT	Trinidad and Tobago
DK	Denmark	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	US	United States of America
FI	Finland	ML	Mali	UZ	Uzbekistan
FR	France	MN	Mongolia	VN	Viet Nam
GA	Gabon				

FLOW CONTROL FOR REAL-TIME DATA STREAMS

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to flow control, i.e., regulation of traffic allowed on a portion of a communications network to avoid excessive congestion.

State of the Art

One of the characteristics of a real-time data stream, such as a videophone data stream, is that it is isochronous--that time is of the essence. If an error occurs in a video or audio stream, the system cannot afford the time to stop everything and retransmit the lost data packets--this will seriously upset the real-time data flow. A better procedure is to just "plow ahead" and pick up the video (or audio) with the next intact frame received.

A similar situation exists with respect to flow control. Known methods of flow control for non-real-time data streams include "stop and wait" flow control and "sliding window" flow control. In stop and wait flow control, a response to data previously sent must be received before any more data may be sent. Stop and wait flow control therefore assumes that the data flow may be interrupted and resumed at will--clearly not the case with real-time data.

In sliding window flow control, flow credits are exchanged and used up. For example, the receiver might allocate a receive buffer of 1000 bytes and send a "send credit" value of 1000 to the sending side. If the sender then sends 100 bytes to the receiver, it keeps track by setting a "sent" variable to 100. At this point the transmitter could send 1000 - 100 = 900 more bytes. As the receiver processes the data and frees up buffer space, it might bump the send credit value to 1000 + 100 = 1100 and send this value to the sending side. The sender would now be allowed to send "send credit" minus "sent" bytes to the receiver, namely 100 - 100 = 1000. As with stop and wait flow control, sliding window

flow control assumes that the data flow may be interrupted and resumed at will. Neither these nor other known methods of flow control are suitable for real-time data streams.

A variety of other approaches to flow control have been proposed, some of which have been implemented. One such technique is *packet discarding*--simply discarding excess packets. Another technique, known as *isarithmic flow control*, limits the total number of packets in the network by using permits that circulate within the network. Whenever a node wants to send a packet, it must first capture a permit and destroy it. The permit is regenerated when the destination node removes the packet from the network. In another approach, which may be referred to as the *choke packet* approach, nodes detecting congestion send "choke packets" back to the source of any message sent into the congested region. The source is then required to reduce or eliminate this type of traffic. Various flow control techniques are described in Grange, J.L., and Gien, M., eds. *Flow Control in Computer Networks*, Amsterdam: North Holland Publishing, 1979.

None of the foregoing flow control mechanisms are optimized for flow control of real-time data streams.

SUMMARY OF THE INVENTION

The present invention, generally speaking, provides for flow control of real-time data streams over computer networks. A real-time data stream is transmitted in data packets from a data source in accordance with a predetermined protocol over a shared network, for example. Data packets of the real-time data stream are received at a data destination connected to the shared network. The data destination determines a suggested data rate for the data source based in part on a number of data packets lost during a preceding interval of time and transmits the suggested data rate to the data source. The suggested data rate is received at the data source, which adjusts its data rate in

accordance with the suggested data rate. The rate adjustment mechanism is designed such that a network segment will not be overloaded with a single isochronous data stream connection and that a disproportionate shared of the network bandwidth is not consumed by the isochronous data stream connection.

BRIEF DESCRIPTION OF THE DRAWING

The present invention may be further understood from the following description in conjunction with the appended drawing. In the drawing:

Figure 1 is a system diagram of computers coupled for exchanging real--time data streams to which the present method of flow control is applicable;

Figure 2A through Figure 2D are diagrams illustrating how a weighted average data rate is determined;

Figure 3A is a graph illustrating how the weighted average data rate is adjusted upward during an error-free run to arrive at a suggested data rate;

Figure 3B is a graph illustrating how the weighted average data rate is adjusted downward during an error-prone run to arrive at a suggested data rate; and

Figure 4 is a graph illustrating the operation of rate averaging.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Figure 1, the present method of flow control is applicable to shared computer networks in which multiple computers 11 exchange real-time data, such as videophone data, across a network 17 including one or more routers such as a router 19. Videophone and other real-time traffic will generate large amounts of data occupying large bandwidths. To prevent videophone users from interfering with other network users, there is provided an automatic means of limiting the data generated (flow control) whenever reduced network performance is detected. In a system composed of multiple network segments connected by route-limiting factors on network throughput is the processing

capacities of the routers themselves: when they are overloaded, they will begin to drop packets. In the following method of flow control, an increase in lost packets is therefore attributed to an overload condition somewhere along the path the videophone packets are travelling. (An increase in lost packets could, of course, be due to a poor quality transmission line along the path--but the receiving node has no way of determining whether such a condition does or does not exist.)

The approach followed by the present method of flow control is to measure the number of packets received from the remote videophone application and the number of packets lost during transmission. As described more fully in U.S. Application Serial No. 08/129,992, filed September 30, 1993, incorporated herein by reference, each packet carries a sequence number whereby the number of packets lost can be readily determined. Based on this information, the receiving node periodically computes a proposed packet transmission rate that will not overload the network. This rate is then sent back to the transmitter (on a reliable control channel, described in the aforementioned copending application), which uses it to adjust parameters under its control to limit its generated bandwidth. These parameters may include the frame rate and compression ratios. If there are multiple streams between the transmitter and receiver, the receiver computes a data rate for each incoming stream but then add the rates for the individual streams and sends the transmitter only the sum for the call. The transmitter is free to reallocate the bandwidth among the various streams of the call as long as the total goal is achieved.

Measurements are made every interval of $\mathbf{T_{goal}}$ ticks (one tick is 1/60 second). For each stream, the number of packets received **P**, the number of bytes received **N** (the total of the lengths of the received packets), and the number of packets lost **L** (determined by counting missing sequence numbers in the stream) during that interval are all recorded.

Computing **L**, the number of lost packets, is complicated by the possibility of packets arriving out of order. The following process **L** accounts for that possibility:

For each stream keep a variable

E = next expected sequence number.

1. If the packet sequence number M == E, set

$$E = E + 1$$
$$P = P + 1$$
$$N = N + \text{(size of packet)}$$

2. If M > E, set

$$L = L + (M - E)$$
$$P = P + 1$$
$$N = N + \text{(size of packet)}$$
$$E = (M + 1)$$

3. If M < E, set

$$L = L - 1$$
$$P = P + 1$$
$$N = N + \text{(size of packet)}$$

The foregoing comparisons may be simplified using modulo arithmetic. For example, if M is in 1/4 of the sequence number space greater than E using modulo arithmetic, then M > E is judged to be true.

If the measurement shows **P** to be zero during the interval, the stream is simply reported to be idle, and none of the following computations are performed.

At the end of each interval of $\mathbf{T}_{actual}$ ticks (the actual elapsed time since the last measurement), the received data rate $\mathbf{R}_{measured}$ (in bytes/second) is computed as follows:

$R_{measured} = (N / T_{actual}) * (60 \text{ ticks/second})$.

The **ErrorRate** (also in bytes/second) is calculated as:

ErrorRate = (L * Rmeasured) / P.

In addition, a variable **ErrorFree** is calculated in accordance with the following procedure:

```
if (L == O)
        ErrorFree++;
else if (L > 1)
        ErrorFree = O;
else if (ErrorFree > O)
        ErrorFree--;
```

The value of **ErrorFree** indicates the number of consecutive intervals that the stream has been error-free. A special case is provided for when a single error is noted during the interval in order to not penalize the flow where there appear to be lost packets but the packets are instead merely out-of-sequence.

To prevent short-term fluctuations in network traffic from causing wild perturbations in data flows, the computed received data rates over the last **B** intervals are tracked (in a vector **r**[1..B]), and averaging is used to smooth out the fluctuations. Rather than giving all the intervals equal weight, most weight is given to the most recent interval (which is **r**[B], **r**[1] being the oldest kept). The weights (which, in a preferred embodiment, must total 100) are kept in a vector **Weights** [1..B].

Next the weighted data rate $\mathbf{R}_{weighted}$ over the last **B** intervals is computed as follows:

$$
\begin{aligned}
&\text{for } (i = 1..B-1) \\
&\qquad r_i = r_i+1 \\
&r_B = R_{measured} \\
&R_{weighted} = (\Sigma_{(i = 1..B)} (W_i \cdot r_i)) / 100
\end{aligned}
$$

The manner in which $\mathbf{R_{weighted}}$ is computed is illustrated in Figures 2A through 2D. The data rates over each of a number intervals are stored in "buckets", each of which has associated with it a weight. In a preferred embodiment, there are four buckets with which the weights 10, 20, 30, and 40 are associated, respectively. The buckets are ordered in a queue such that the data rate over the most recent interval is stored in the bucket having a weight of 40, the data rate over the next-most-recent interval is stored in the bucket having a weight of 30, etc.

Referring particularly to Figure 2A, after the data rate during a first interval has been measured, all of the buckets are initialized to this value, for example 752, meaning that 752 bytes were received during the interval. Assume that 824 bytes are received during the next interval. At the end of the next interval the value 824 replace the value 752 in the last bucket as shown in Figure 2B. Assume now that 511 bytes are received during the next interval and that 696 bytes are received during the next interval after that. These values are stored in the buckets at the conclusion of the respective intervals as illustrated in Figures 2C and 2D. At the conclusion of the fourth interval, the weighted average is 671 bytes, represented by a straight line having an area underneath equal to the area under the line segments representing the "levels" of the respective buckets.

Once $\mathbf{R_{weighted}}$ has been computed, a suggested data rate $\mathbf{R_{suggested}}$ to be used by the remote transmitter may then be computed. However, in order to dampen wild fluctuations that may occur when operating under some operating systems, a variable **Cap** is used to limit the maximum change in $\mathbf{R_{suggested}}$ based upon the previous value of $\mathbf{R_{suggested}}$ as follows:

Cap = $R_{suggested} \cdot 25\%$.

If the flow is experiencing a good run of error-free intervals, the suggested data rate is boosted, as a reward, by a value that increases for each consecutive error-free interval, but not by more than a certain percentage (**ErrorFreeCap**) of a maximum allowed rate **MaxRate**. Conversely, when errors are encountered, the suggested data rate is penalized by more, by a factor **ErrorRateFactor**, than the measured **ErrorRate** as follows:

```
if (ErrorFree > O)
        R_suggested = R_weighted
          + min ((MaxRate * min(ErrorFree, ErrorFreeCap))
/ 100, Cap)
else
        R_suggested = R_weighted
          - min ((ErrorRate * ErrorRateFactor), Cap)
```

The manner in which $\mathbf{R_{weighted}}$ is adjusted to arrive at $\mathbf{R_{suggested}}$ is illustrated in Figures 3A and 3B.

Referring particularly to Figure 3A, the weighted average data rate is adjusted upward during an error-free run to arrive at a suggested data rate. In a preferred embodiment, for each consecutive error-free interval up to the maximum **ErrorFreeCap**, the weighted average data rate $\mathbf{R_{weighted}}$ is increased by one percent of the maximum allowed rate **MaxRate**. This increase is subject in addition to the limit **Cap**, which in a preferred embodiment is based on the previous value of $\mathbf{R_{suggested}}$ and be greater than or less than the cap computed using **ErrorFreeCap**.

Referring to Figure 3B, the weighted average data rate is adjusted downward during an error-prone run to arrive at a suggested data rate. The weighted average data rate is decreased by an amount that increases linearly with **ErrorRate**. The constant **ErrorRateFactor** is the slope of the line representing

$R_{suggested}$ and is the proportionality constant between $R_{suggested}$ and **ErrorRate**. The decrease in the weighted average data rate is subject to the same cap as in Figure 3A.

Finally, after computing the $R_{suggested}$ values for all streams on the call, they are added together and the total sent to the transmitter over the reliable control channel. The transmitter then is required to adjust its data generation rate accordingly.

In a preferred embodiment measurement intervals are not synchronized with the flow. When starting up from the idle state, the data from the first measurement interval is therefore indubitably wrong, inasmuch as the measurement does not cover the complete interval. The data from the first measurement interval is therefore discarded. The computations begin instead using, the second interval data by initializing **ErrorFree** to zero. $R_{suggested}$ to **MaxRate**, and filling all values of the vector **r** with the first computed $R_{measured}$ before computing the first $R_{weighted}$.

If the transmitter starts off transmitting immediately at **MaxRate**, it is quite possible to overload particularly slow routers with so much traffic that even the first value of $R_{suggested}$ cannot be returned from the far end returned reliably. To avoid this possibility, transmitters are required to start transmitting on a newly minted stream at a lower rate **InitialRate**. As experience is accumulated this rate is adjusted up or down.

In an exemplary embodiment the following values of the foregoing global constants were used:

T_{goal}	= (5 seconds) $\cdot$ (60 ticks/second) = 300 ticks
B	= 4 buckets
Weights	= {10, 20, 30, 40}
MaxRate	= ((6,000,000 bps) / 2) / (8 bits/byte) = 375,000 bytes/second

Initial Rate	= MaxRate * 25%
ErrorFreeCap	= 7
ErrorRateFactor	= 10

In addition, the following variables are kept per active stream:

r[1..B]	= rate history
ErrorFree	= count of error-free intervals
$R_{suggested}$	= data rate the remote transmitter should use

One characteristic of the described method of flow control is that once a transmitter has succeeded in obtaining bandwidth, it will tend to keep that bandwidth in preference to later attempts by other transmitters to obtain bandwidth. So, for example, if one end of a call is put on hold temporarily, the other end will gradually absorb and attempt to keep the vacated bandwidth, making it hard for the one end to regain bandwidth when taken off hold.

One solution to this problem is rate averaging, which may be performed by the transmitter when it detects the rate it is allowed to transmit, the $\mathbf{R}_{suggested}$ just received, is larger than the suggested rate the transmitter is sending to the other side. Since the data flowing in both directions is likely to be traveling the same path, the transmitter and the receiver are both benefited by cooperating to allocate the channel bandwidth. When the foregoing condition is detected, the transmitter therefore voluntarily lower its transmission rate to halfway between the two $\mathbf{R}_{suggested}$ values to approach a more equitable distribution.

This rate averaging method also compensates for possible disparities in the capabilities of the two machines involved in the transmission one of which may be capable of seizing a disproportionate amount of the bandwidth.

The manner in which rate averaging achieves a more equitable bandwidth distribution is illustrated in Figure 4. As the "dominant" station voluntarily reduces its transmission rate, the opposite station reduces the

suggested data rate that it sends to the dominant station. At the same time, the "subservient" station is able to seize more bandwidth as the dominant station vacates bandwidth. The suggested and actual data rates of the two stations therefore tend to converge.

The maximum data rate as described above is set in an attempt to not overload an Ethernet segment with a single videophone or other real-time data call.

The foregoing has described the principles, preferred embodiments, and modes of operation of the present invention. However, the invention should not be limited to the embodiments discussed. The above-described embodiments should therefore be regarded as illustrative rather than restrictive. Variations in those embodiments may be made without departing from the scope of the present invention as defined by the following claims. For example, the measurement interval length may be adjusted. Lowering the measurement interval length will cause the flow control process respond more quickly to network changes. The number of buckets used may be increased or decreased to keep a longer or shorter history of network traffic. The weights used may be changed, for example to make them less exponential. Furthermore, rather than calculating the suggested data rate at the receiver, the receiver may simply send raw measurements to the transmitter from which the transmitter may then calculated the suggested data rate. Other variations will be apparent to one of ordinary skill in the art.

What is claimed is:

1. A method comprising:

transmitting in data packets in accordance with a predetermined protocol a real-time data stream from a data source over a shared computer network;

receiving data packets of said real-time data stream at a data destination connected to said shared network;

determining a suggested data rate for the data source based in part on a number of lost data packets transmitted by the data source but not received by the data destination during a preceding interval of time; and

transmitting from the data destination to the data source information related to the suggested data rate.

2. The method of Claim 1, wherein the suggested data rate is determined at the data destination and transmitted from the data destination to the data source.

3. The method of Claim 2, comprising the further steps of receiving the suggested data rate at the data source; and adjusting a data rate of the data source in accordance with the suggested data rate.

4. The method of Claim 3, wherein the suggested data rate is transmitted across a reliable control channel

5. The method of Claim 1, wherein the suggested data rate is determined at the data source using the information transmitted from the data destination to the data source.

6. The method of Claim 5, comprising the further step of:

adjusting a data rate of the data source in accordance with the suggested data rate.

7. The method of Claim 1, wherein the determining step comprises:
measuring a data rate of the real-time data stream during respective ones of a plurality of substantially equal preceding intervals of time.

8. The method of Claim 7, wherein the determining step further comprises:
forming an average of the data rate of the real-time data stream during a plurality of intervals.

9. The method of Claim 8, wherein the average is a weighted average of the data rate of the real-time data stream during a plurality of intervals.

10. The method of Claim 9, wherein the average is a weighted average of the data rate of the real-time data stream during each of a predetermined number of consecutive preceding intervals.

11. The method of Claim 8, wherein the suggested data rate is determined by adjusting the average data rate of the real-time data stream during a plurality of intervals in accordance with the number of lost data packets transmitted by the data source but not received by the data destination during a preceding interval of time.

12. The method of Claim 11, wherein the suggested data rate is determined by adding an increment to the average data rate when the number of lost packets during a preceding interval of time was low.

13. The method of Claim 12, wherein an increment is added to the average data rate when the number of lost packets during an immediately preceding interval of time was zero.

14. The method of Claim 13, wherein a greater increment is added to the average data rate as a number of preceding intervals in which the number of packets lost was zero increases.

15. The method of Claim 14, wherein the preceding intervals in which the number of packets lost was zero are consecutive intervals.

16. The method of Claim 14, wherein the size of increment added to the average data rate is subject to an upper limit.

17. The method of Claim 11, wherein an increment is subtracted from the average data rate when the number of lost packets during an immediately preceding time interval was non-zero.

18. The method of Claim 17, wherein a greater increment is subtracted from the average data rate as a proportion of packets lost to packets received increases.

19. The method of Claim 18, wherein the size of the increment subtracted from the average data rate is subject to an upper limit.

20. The method of Claim 11, wherein, at start-up, the real-time data stream is transmitted at an initial data rate lower than a maximum allowed data rate.

21. The method of Claim 11, wherein, at the conclusion of a first interval following start-up, a proportion of packets lost to packets received during the first interval is ignored when determining the suggested data rate at the conclusion of the second and subsequent intervals.

22. A method comprising:

transmitting in data packets in accordance with a predetermined protocol a plurality of real-time data streams from a data source over a shared computer network;

receiving data packets of said plurality of real-time data streams at a data destination connected to said shared network;

determining a suggested data rate for the each of the plurality of real-time data streams based in part on a number of lost data packets transmitted by the data source but not received by the data destination in each during a preceding interval of time;

adding together suggested data rates for each of the plurality of real-time data streams to arrive at an aggregate suggested data rate;

transmitting from the data destination to the data source information related to the aggregate suggested data rate; and

adjusting data rates of the plurality of real-time data streams at the data source such that a combined data rate of the plurality of real-time data streams does not exceed the aggregate suggested data rate.

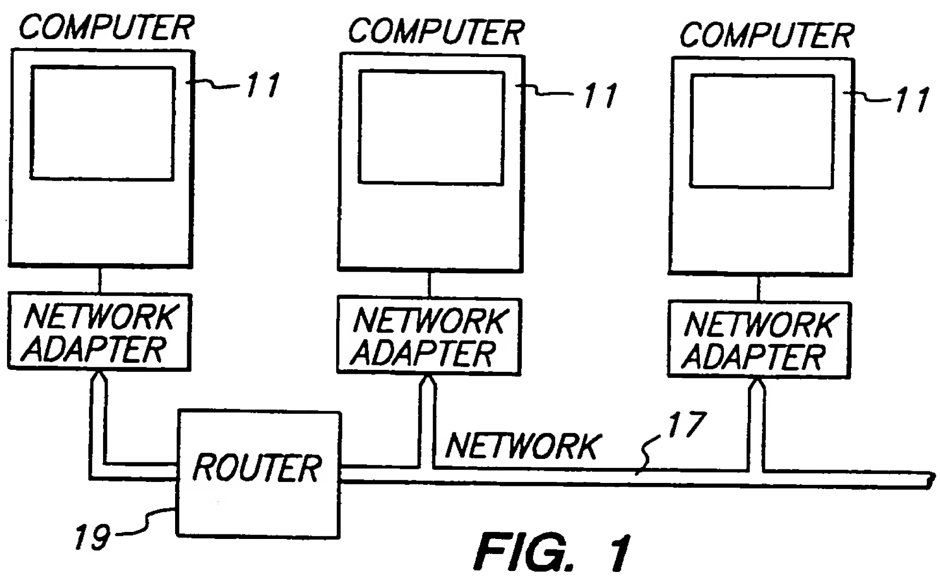


FIG. 1

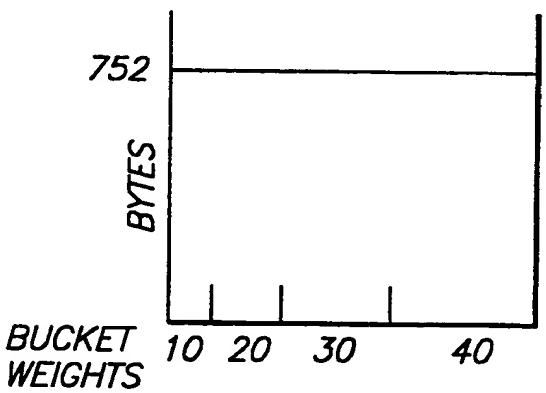


FIG. 2A

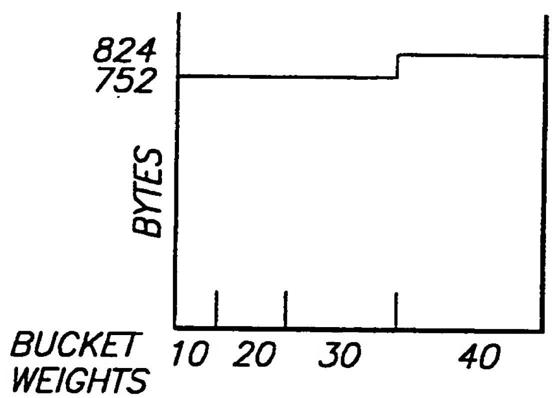


FIG. 2B

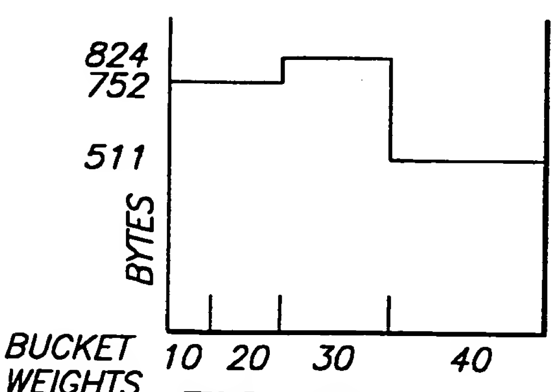


FIG. 2C

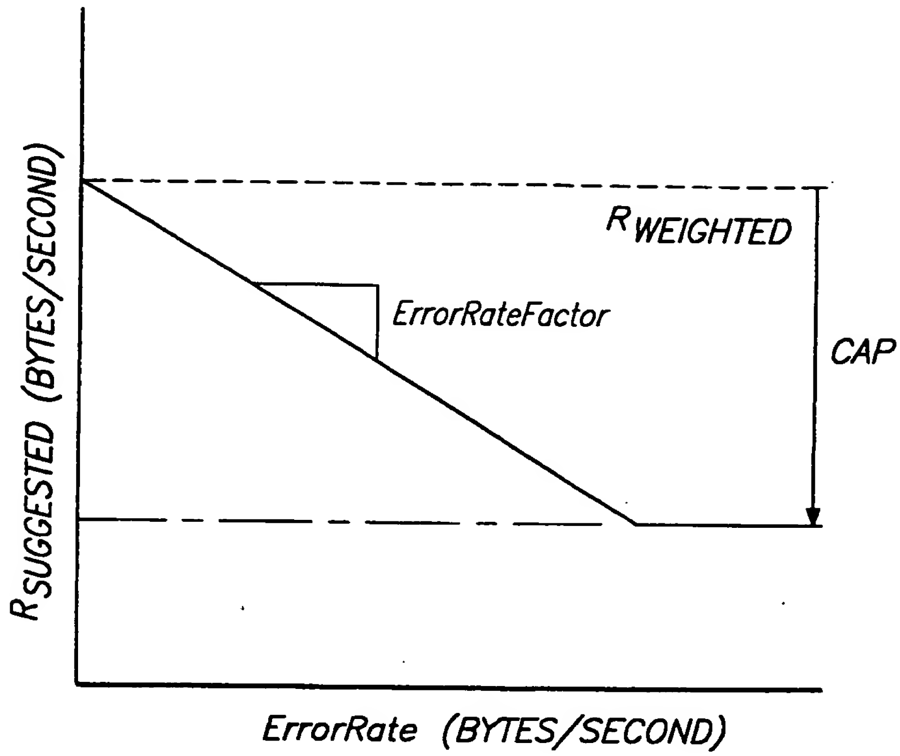

R WEIGHTED
ErrorRateFactor
CAP
R SUGGESTED (BYTES/SECOND)
ErrorRate (BYTES/SECOND)

FIG. 3B

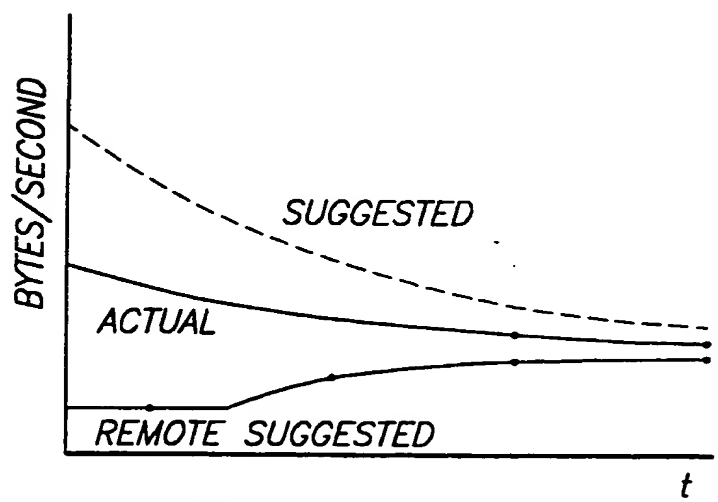

BYTES/SECOND
SUGGESTED
ACTUAL
REMOTE SUGGESTED
t

FIG. 4

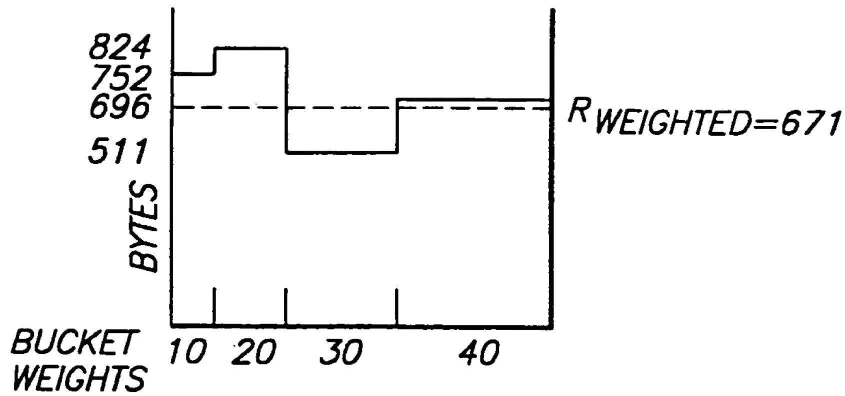

824
752
696
511
BYTES
$R_{WEIGHTED}=671$
BUCKET WEIGHTS
10
20
30
40

FIG. 2D

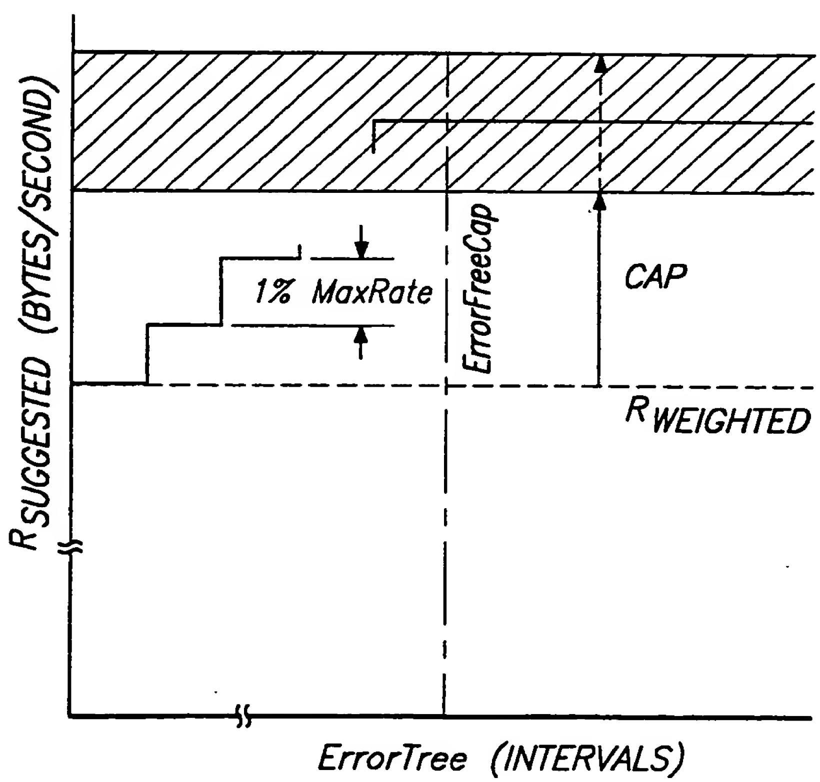

$R_{SUGGESTED}$ (BYTES/SECOND)
1% MaxRate
ErrorFreeCap
CAP
$R_{WEIGHTED}$
ErrorTree (INTERVALS)

FIG. 3A

INTERNATIONAL SEARCH REPORT

Intern   ial Application No
PCT/US 95/04746

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP-A-0 446 956 (FUJITSU LIMITED) 18 September 1991	1-3,5,6
Y		4,7
	see claim 9	
A		22

Y	IEEE / ACM TRANSACTIONS ON NETWORKING, vol. 1, no. 6, December 1993 NEW YORK US, pages 693-708, L. BENMOHAMED 'Feedback control of congestion in packet switching networks: the case of a single congested node' see paragraph 2.1.2	4,7
A		1,8-11

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
31 August 1995	18.09.95

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Perez Perez, J

Form PCT/ISA/210 (second sheet) (July 1992)

Intern· al Application No
PCT/US 95/04746

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SIGCOMM '91 CONFERENCE, September 1991 USA, pages 17-28, C.L. WILLIAMSON ET AL. 'Loss-load curver: support for-based congestion control in high-speed datagram networks' see paragraph 2.3 ---	1,7-10, 22
A	PHOENIX CONFERENCE ON COMPUTERS AND COMMUNICATIONS, March 1992 USA, pages 315-322, Z WANG ET AL. 'A fluid model approximation to quantitative information feedback in congestion control' see paragraph 6 -----	1-22

1

INTERNATIONAL SEARCH REPORT

..formation on patent family members

Intern al Application No
PCT/US 95/04746

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-446956	18-09-91	JP-A- 3267846	28-11-91